# SPYWOLF

## Security Audit Report

Completed on
**March 30, 2023**

SPYWOLF.CO
@SPYWOLFNETWORK
@SPYWOLFNETWORK

# OVERVIEW

This audit has been prepared for **DogAI** to review the main aspects of the project to help investors make make an informative decision during their research process.

You will find a a summarized review of the following key points:

- ✓ Contract's source code
- ✓ Owners' wallets
- ✓ Tokenomics
- ✓ Team transparency and goals
- ✓ Website's age, code, security and UX
- ✓ Whitepaper and roadmap
- ✓ Social media & online presence

> *The results of this audit are purely based on the team’s evaluation and does not guarantee nor reflect the projects outcome and goal*
>
> - SPYWOLF Team -

# TABLE OF CONTENTS

Project Description 01

Contract 1 (Flexible Stake) 02

- Current Stats 03
- Vulnerability Check 04
- Found Threats 06-A/06-C

Contract 2 (Locked - 6 Months) 07

- Current Stats 08
- Vulnerability Check 09
- Found Threats 10-A/10-C

Contract 3 (Locked - 12 Months) 11

- Current Stats 12
- Vulnerability Check 13
- Found Threats 14-A/14-C

About SPYWOLF 15

Disclaimer 16

# DOGAI

## PROJECT DESCRIPTION

**According to their whitepaper:**
DogAI is a fully decentralized and self-managed protocol that enables their investors to easily and securely interact with their tokens, supported by superior risk analysis and smart contract management tools. They are creating a new ecosystem of automated digital asset management that utilizes AI with analytical data and makes it available to users.

**Release Date:** Launched on February 23, 2023

**Category:** AI

# CONTRACT 1 FLEXIBLE STAKE

|   |   |
|---|---|
| Token Name<br>Flexible Stake | Symbol<br>N/A |
| Contract Address<br>0x30A357Fb737C7De6e353cD8d4CFf848B49B2d747 | |
| Network<br>Binance Smart Chain | Language<br>Solidity |
| Deployment Date<br>March 13, 2023 | Verified?<br>Yes |
| Total Supply<br>N/A | Status<br>Deployed |

# TAXES

| Buy Tax | Sell Tax |
|---|---|
| **none** | **none** |

*Taxes cannot be changed in future

# Our Contract Review Process

The contract review process pays special attention to the following:

- ✓ Testing the smart contracts against both common and uncommon vulnerabilities
- ✓ Assessing the codebase to ensure compliance with current best practices and industry standards.
- ✓ Ensuring contract logic meets the specifications and intentions of the client.
- ✓ Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- ✓ Thorough line-by-line manual review of the entire codebase by industry experts.

**Blockchain security tools used:**

- OpenZeppelin
- Mythril
- Solidity Compiler
- Hardhat

## TOKEN TRANSFERS STATS

| | |
|---|---|
| **Transfer Count** | N/A |
| **Uniq Senders** | N/A |
| **Uniq Receivers** | N/A |
| **Total Amount** | N/A |
| **Median Transfer Amount** | N/A |
| **Average Transfer Amount** | N/A |
| **First transfer date** | N/A |
| **Last transfer date** | N/A |
| **Days token transferred** | N/A |

## SMART CONTRACT STATS

| | |
|---|---|
| **Calls Count** | 2 |
| **External calls** | 2 |
| **Internal calls** | 0 |
| **Transactions count** | 2 |
| **Uniq Callers** | 1 |
| **Days contract called** | 3 |
| **Last transaction time** | Mar-16-2023 04:55:45 PM +UTC |
| **Created** | Mar-13-2023 04:21:41 PM +UTC |
| **Create TX** | 0xb64ef91667a5b2cbe1bfea33a30bcf4a0f3a56d594017acdb1424d72d4d87d8e |
| **Creator** | 0xdb7ca7384d0019514a609a606e7bee73af4751eb |

# VULNERABILITY CHECK

| | |
|---|---|
| Design Logic | Passed |
| Compiler warnings. | Passed |
| Private user data leaks | Passed |
| Timestamp dependence | Passed |
| Integer overflow and underflow | Passed |
| Race conditions and reentrancy. Cross-function race conditions | Passed |
| Possible delays in data delivery | Passed |
| Oracle calls | Passed |
| Front running | Passed |
| DoS with Revert | Passed |
| DoS with block gas limit | Passed |
| Methods execution permissions | Passed |
| Economy model | Passed |
| Impact of the exchange rate on the logic | Passed |
| Malicious Event log | Passed |
| Scoping and declarations | Passed |
| Uninitialized storage pointers | Passed |
| Arithmetic accuracy | Passed |
| Cross-function race conditions | Passed |
| Safe Zeppelin module | Passed |
| Fallback function security | Passed |

# THREAT LEVELS

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and access control issues within the code. The exploitation of these issues by malicious actors may cause serious financial damage to projects that failed to get an audit in time. We categorize these vulnerabilities by the following levels:

## High Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

## Medium Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

## Low Risk

Issues on this level are minor details and warning that can remain unfixed.

## Informational

Information level is to offer suggestions for improvement of efficacy or security for features with a risk free factor.

# FOUND THREATS

## ⚠ High Risk

No high risk-level threats found in this contract.

## ⚠ Medium Risk

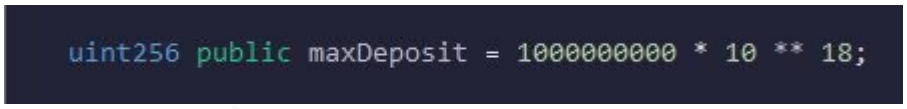
No medium risk-level threats found in this contract.

## ⚠ Low Risk

No low risk-level threats found in this contract.

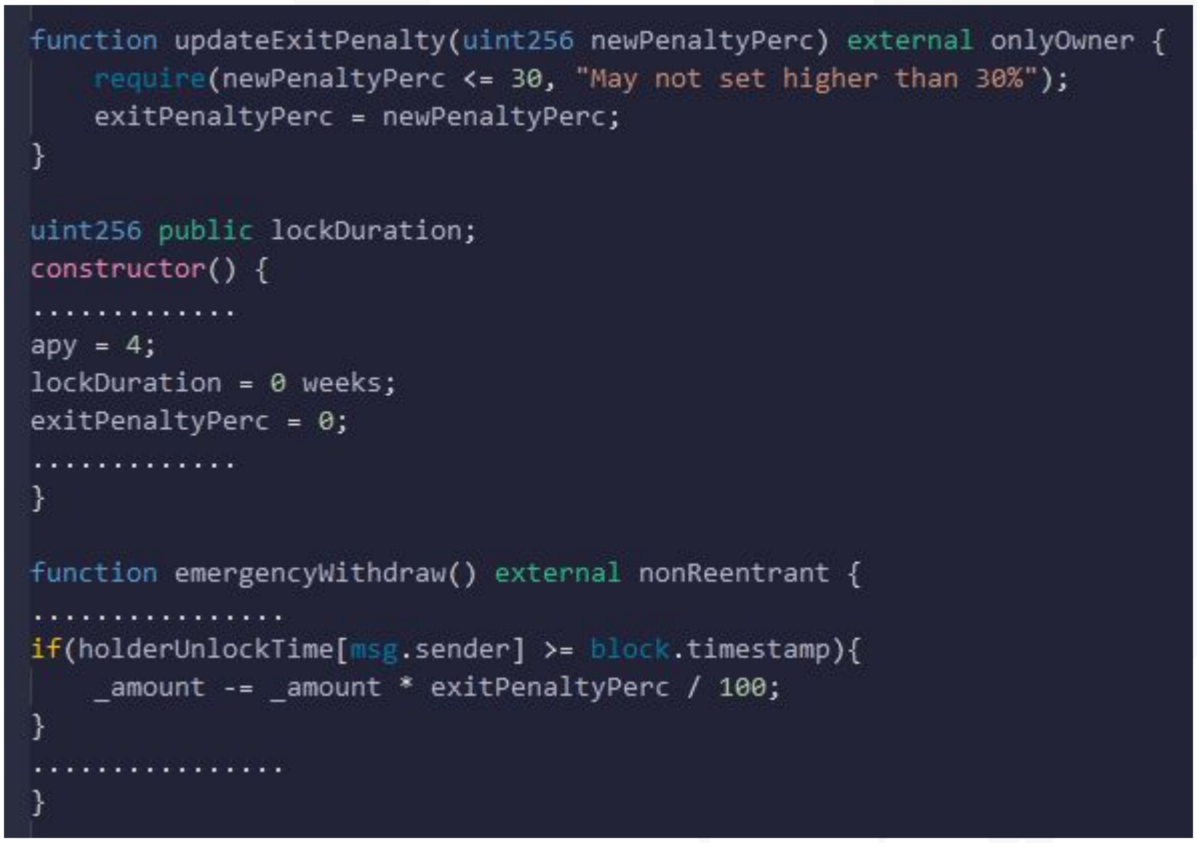
# Informational

Maximum stake per user for this contract is 1,000,000,000 $DOGAI.



```
uint256 public maxDeposit = 1000000000 * 10 ** 18;
```

Owner can set emergency withdraw fee up to 30% if user withdraw earlier than the LockDuration  period.
Current LockDuration is 0 (same as user deposit time) and cannot be changed. The penalty fees will not apply even if they are set above 0.
Users can withdraw without any penalty fees applied at any time.
Current APY is 4%.



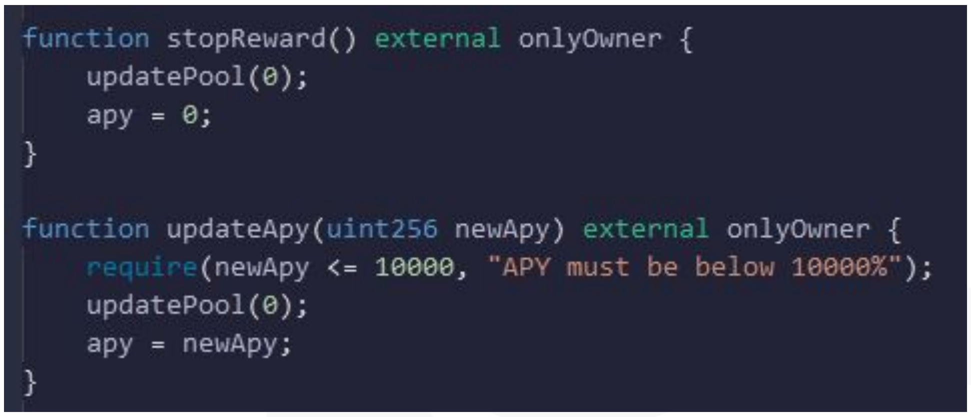
```
function updateExitPenalty(uint256 newPenaltyPerc) external onlyOwner {
    require(newPenaltyPerc <= 30, "May not set higher than 30%");
    exitPenaltyPerc = newPenaltyPerc;
}

uint256 public lockDuration;
constructor() {
............
apy = 4;
lockDuration = 0 weeks;
exitPenaltyPerc = 0;
............
}

function emergencyWithdraw() external nonReentrant {
................
if(holderUnlockTime[msg.sender] >= block.timestamp){
    _amount -= _amount * exitPenaltyPerc / 100;
}
................
}
```

# Informational

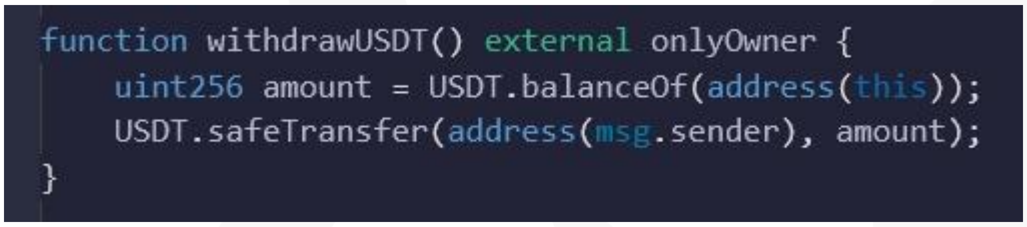
Owner can set APY to from 0 to 10000%.
When APY is set at 0, no staking rewards will be paid.



```
function stopReward() external onlyOwner {
    updatePool(0);
    apy = 0;
}

function updateApy(uint256 newApy) external onlyOwner {
    require(newApy <= 10000, "APY must be below 10000%");
    updatePool(0);
    apy = newApy;
}
```

Owner can withdraw USDT tokens from the contract.



```
function withdrawUSDT() external onlyOwner {
    uint256 amount = USDT.balanceOf(address(this));
    USDT.safeTransfer(address(msg.sender), amount);
}
```

# CONTRACT 2
## LOCKED (6 MONTHS)

| | |
|---|---|
| Token Name<br>LockedSixStake | Symbol<br>N/A |

Contract Address
0xD4267b16612dD09C87A16267eEdAaa69E1f7f9Ab

| | |
|---|---|
| Network<br>Binance Smart Chain | Language<br>Solidity |
| Deployment Date<br>March 13, 2023 | Verified?<br>Yes |
| Total Supply<br>N/A | Status<br>Not launched |

## TAXES

| Buy Tax | Sell Tax |
|---|---|
| none | 10% |

*Taxes can be changed in future

# Our Contract Review Process

The contract review process pays special attention to the following:

- ✓ Testing the smart contracts against both common and uncommon vulnerabilities
- ✓ Assessing the codebase to ensure compliance with current best practices and industry standards.
- ✓ Ensuring contract logic meets the specifications and intentions of the client.
- ✓ Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- ✓ Thorough line-by-line manual review of the entire codebase by industry experts.

**Blockchain security tools used:**

- OpenZeppelin
- Mythril
- Solidity Compiler
- Hardhat

# TOKEN TRANSFERS STATS

| | |
|---|---|
| **Transfer Count** | N/A |
| **Uniq Senders** | N/A |
| **Uniq Receivers** | N/A |
| **Total Amount** | N/A |
| **Median Transfer Amount** | N/A |
| **Average Transfer Amount** | N/A |
| **First transfer date** | N/A |
| **Last transfer date** | N/A |
| **Days token transferred** | N/A |

# SMART CONTRACT STATS

| | |
|---|---|
| **Calls Count** | 2 |
| **External calls** | 2 |
| **Internal calls** | 0 |
| **Transactions count** | 2 |
| **Uniq Callers** | 1 |
| **Days contract called** | 3 |
| **Last transaction time** | Mar-16-2023 04:56:06 PM +UTC |
| **Created** | Mar-13-2023 04:23:14 PM +UTC |
| **Create TX** | 0x91cbd5b146cea99e1161f9071e4ed64de8e706133ad915c361c3ee7bb81beace |
| **Creator** | 0xdb7ca7384d0019514a609a606e7bee73af4751eb |

# VULNERABILITY CHECK

| | |
|---|---|
| Design Logic | Passed |
| Compiler warnings. | Passed |
| Private user data leaks | Passed |
| Timestamp dependence | Passed |
| Integer overflow and underflow | Passed |
| Race conditions and reentrancy. Cross-function race conditions | Passed |
| Possible delays in data delivery | Passed |
| Oracle calls | Passed |
| Front running | Passed |
| DoS with Revert | Passed |
| DoS with block gas limit | Passed |
| Methods execution permissions | Passed |
| Economy model | Passed |
| Impact of the exchange rate on the logic | Passed |
| Malicious Event log | Passed |
| Scoping and declarations | Passed |
| Uninitialized storage pointers | Passed |
| Arithmetic accuracy | Passed |
| Cross-function race conditions | Passed |
| Safe Zeppelin module | Passed |
| Fallback function security | Passed |

# FOUND THREATS

## High Risk

No high risk-level threats found in this contract.

## Medium Risk

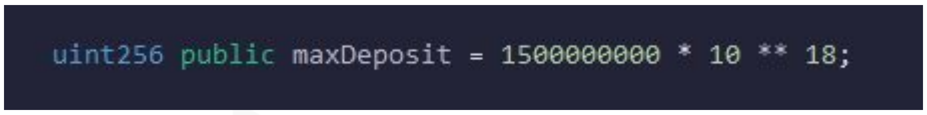
No medium risk-level threats found in this contract.

## Low Risk

No low risk-level threats found in this contract.

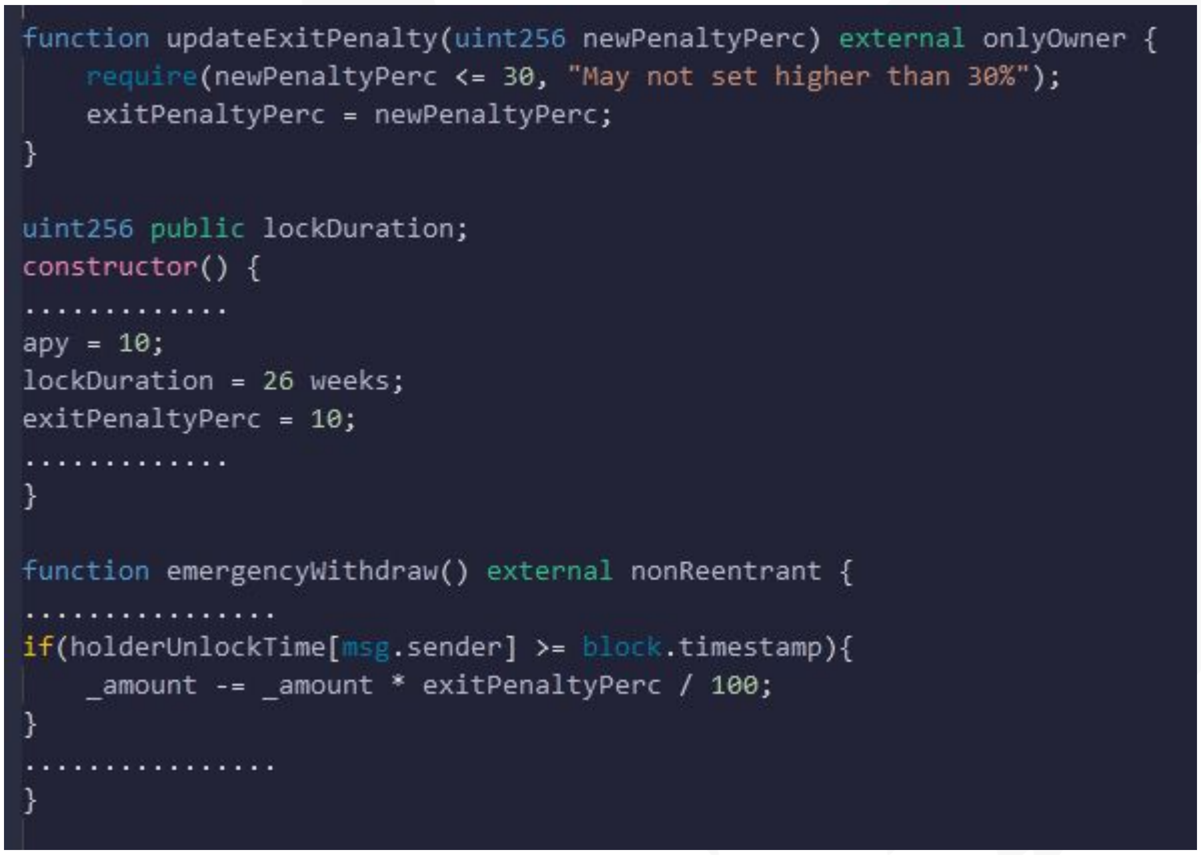
# Informational

Maximum stake per user for this contract is 1,500,000,000  $DOGAI.



```
uint256 public maxDeposit = 1500000000 * 10 ** 18;
```

Owner can set emergency withdraw fee up to 30% if user withdraw earlier than the LockDuration  period.
Current LockDuration period is 26 weeks and cannot be changed.
Current early withdraw penalty fees are 10%.
Current APY is 10%.



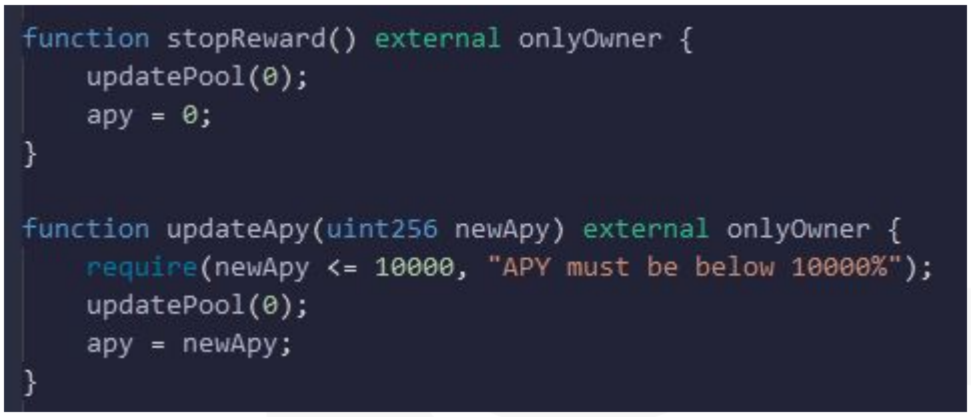
```
function updateExitPenalty(uint256 newPenaltyPerc) external onlyOwner {
    require(newPenaltyPerc <= 30, "May not set higher than 30%");
    exitPenaltyPerc = newPenaltyPerc;
}

uint256 public lockDuration;
constructor() {
.............
apy = 10;
lockDuration = 26 weeks;
exitPenaltyPerc = 10;
.............
}

function emergencyWithdraw() external nonReentrant {
................
if(holderUnlockTime[msg.sender] >= block.timestamp){
    _amount -= _amount * exitPenaltyPerc / 100;
}
................
}
```

# Informational

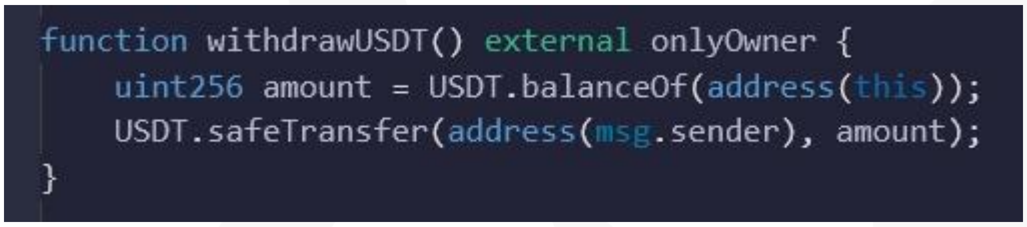
Owner can set APY to from 0 to 10000%.
When APY is set at 0, no staking rewards will be paid.



```
function stopReward() external onlyOwner {
    updatePool(0);
    apy = 0;
}

function updateApy(uint256 newApy) external onlyOwner {
    require(newApy <= 10000, "APY must be below 10000%");
    updatePool(0);
    apy = newApy;
}
```

Owner can withdraw USDT tokens from the contract.



```
function withdrawUSDT() external onlyOwner {
    uint256 amount = USDT.balanceOf(address(this));
    USDT.safeTransfer(address(msg.sender), amount);
}
```

# CONTRACT 3
## LOCKED (12 MONTHS)

Token Name
LockedTwelveStake

Symbol
N/A

Contract Address
0x23C8c8f641334Ba670c95A6E83d0442781156d92

Network
Binance Smart Chain

Language
Solidity

Deployment Date
March 13, 2023

Verified?
Yes

Total Supply
N/A

Status
Not launched

## TAXES

Buy Tax
**none**

Sell Tax
**10%**

*Taxes can be changed in future

# Our Contract Review Process

The contract review process pays special attention to the following:

- ✓ Testing the smart contracts against both common and uncommon vulnerabilities
- ✓ Assessing the codebase to ensure compliance with current best practices and industry standards.
- ✓ Ensuring contract logic meets the specifications and intentions of the client.
- ✓ Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- ✓ Thorough line-by-line manual review of the entire codebase by industry experts.

**Blockchain security tools used:**

- OpenZeppelin
- Mythril
- Solidity Compiler
- Hardhat

# TOKEN TRANSFERS STATS

| | |
|---|---|
| **Transfer Count** | N/A |
| **Uniq Senders** | N/A |
| **Uniq Receivers** | N/A |
| **Total Amount** | N/A |
| **Median Transfer Amount** | N/A |
| **Average Transfer Amount** | N/A |
| **First transfer date** | N/A |
| **Last transfer date** | N/A |
| **Days token transferred** | N/A |

# SMART CONTRACT STATS

| | |
|---|---|
| **Calls Count** | 2 |
| **External calls** | 2 |
| **Internal calls** | 0 |
| **Transactions count** | 2 |
| **Uniq Callers** | 1 |
| **Days contract called** | 3 |
| **Last transaction time** | Mar-16-2023 04:56:27 PM +UTC |
| **Created** | Mar-13-2023 04:25:35 PM +UTC |
| **Create TX** | 0x70636f6afb7cef7f6fdaacb403ec9d3381f942048ae74930c33100eda9814e4d |
| **Creator** | 0xdb7ca7384d0019514a609a606e7bee73af4751eb |

# VULNERABILITY CHECK

| | |
|---|---|
| Design Logic | Passed |
| Compiler warnings. | Passed |
| Private user data leaks | Passed |
| Timestamp dependence | Passed |
| Integer overflow and underflow | Passed |
| Race conditions and reentrancy. Cross-function race conditions | Passed |
| Possible delays in data delivery | Passed |
| Oracle calls | Passed |
| Front running | Passed |
| DoS with Revert | Passed |
| DoS with block gas limit | Passed |
| Methods execution permissions | Passed |
| Economy model | Passed |
| Impact of the exchange rate on the logic | Passed |
| Malicious Event log | Passed |
| Scoping and declarations | Passed |
| Uninitialized storage pointers | Passed |
| Arithmetic accuracy | Passed |
| Cross-function race conditions | Passed |
| Safe Zeppelin module | Passed |
| Fallback function security | Passed |

# FOUND THREATS

## High Risk

No high risk-level threats found in this contract.

## Medium Risk

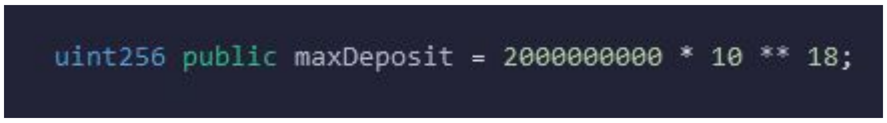
No medium risk-level threats found in this contract.

## Low Risk

No low risk-level threats found in this contract.

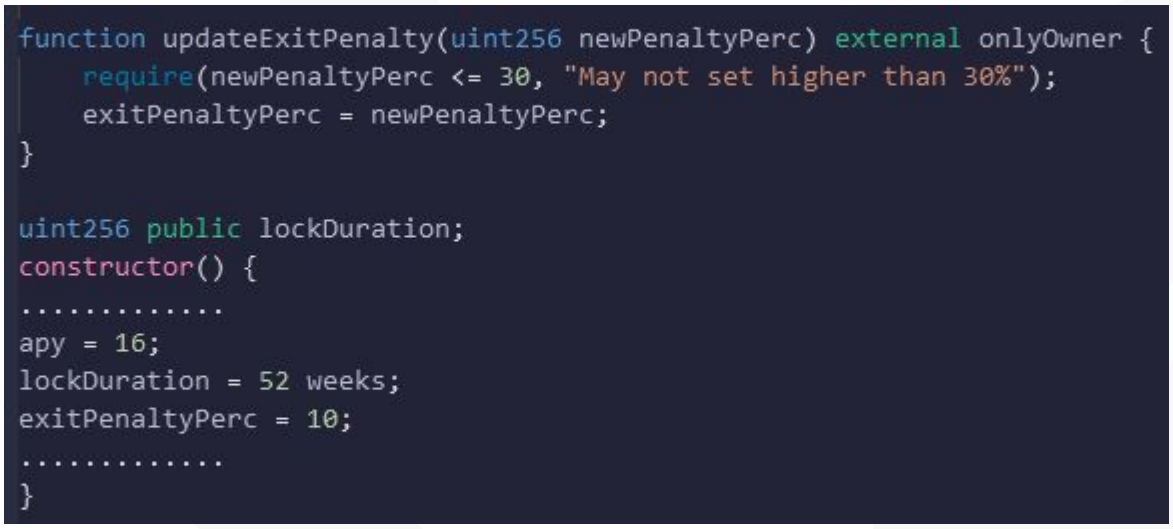
# Informational

Maximum stake per user for this contract is 2,000,000,000  $DOGAI.



```
uint256 public maxDeposit = 2000000000 * 10 ** 18;
```

Owner can set emergency withdraw fee up to 30% if user withdraw earlier than the LockDuration  period.
Current LockDuration period is 52 weeks and cannot be changed.
Current early withdraw penalty fees are 10%.
Current APY is 16%.

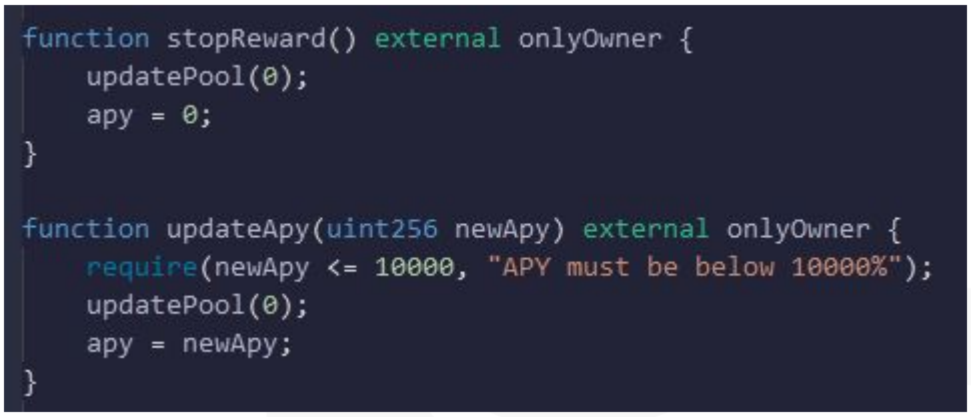


```
function updateExitPenalty(uint256 newPenaltyPerc) external onlyOwner {
    require(newPenaltyPerc <= 30, "May not set higher than 30%");
    exitPenaltyPerc = newPenaltyPerc;
}

uint256 public lockDuration;
constructor() {
.............
apy = 16;
lockDuration = 52 weeks;
exitPenaltyPerc = 10;
.............
}
```

# Informational

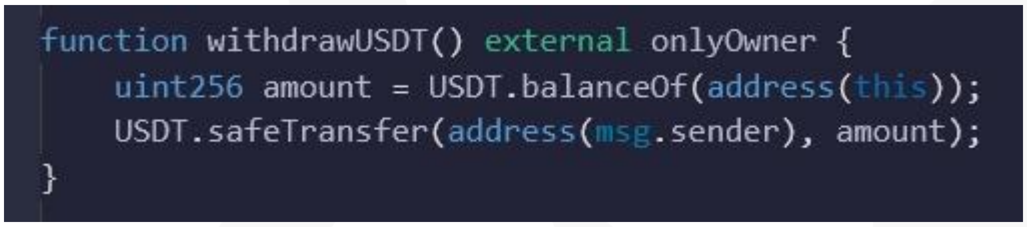
Owner can set APY to from 0 to 10000%.
When APY is set at 0, no staking rewards will be paid.



```
function stopReward() external onlyOwner {
    updatePool(0);
    apy = 0;
}

function updateApy(uint256 newApy) external onlyOwner {
    require(newApy <= 10000, "APY must be below 10000%");
    updatePool(0);
    apy = newApy;
}
```

Owner can withdraw USDT tokens from the contract.



```
function withdrawUSDT() external onlyOwner {
    uint256 amount = USDT.balanceOf(address(this));
    USDT.safeTransfer(address(msg.sender), amount);
}
```

SPYWOLF

CRYPTO SECURITY

Audits | KYCs | dApps
Contract Development

# ABOUT US

We are a growing crypto security agency offering audits, KYCs and consulting services for some of the top names in the crypto industry.

- ✓ OVER 400 SUCCESSFUL CLIENTS
- ✓ MORE THAN 500 SCAMS EXPOSED
- ✓ MILLIONS SAVED IN POTENTIAL FRAUD
- ✓ PARTNERSHIPS WITH TOP LAUNCHPADS, INFLUENCERS AND CRYPTO PROJECTS
- ✓ CONSTANTLY BUILDING TOOLS TO HELP INVESTORS DO BETTER RESEARCH

To hire us, reach out to
contact@spywolf.co or
t.me/joe_SpyWolf

## FIND US ONLINE

SPYWOLF.CO

@SPYWOLFNETWORK

@SPYWOLFNETWORK

# Disclaimer

This report shows findings based on our limited project analysis, following good industry practice from the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, overall social media and website presence and team transparency details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report.

While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

**DISCLAIMER:**

By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice.

No one shall have any right to rely on the report or its contents, and SpyWolf and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (SpyWolf) owe no duty of care towards you or any other person, nor does SpyWolf make any warranty or representation to any person on the accuracy or completeness of the report.

The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and SpyWolf hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, SpyWolf hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against SpyWolf, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts, website, social media and team.

No applications were reviewed for security. No product code has been reviewed.